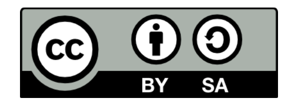
Git
Fundamentos

Guion de comandos



Git. Fundamentos. Guion de comandos

Desarrollado por Jesús Amieiro Becerra



Versión actual: 27/10/2015

Puedes obtener la última versión disponible en http://fontelearn.com/es/git-fundamentos/ o en http://www.jesusamieiro.com/docs/

Índice de contenido

1 Instalación

1.1 Windows

```
# Instalación en Windows
# Instalación mediante interfaz gráfica
pwd
ls -la
git --version
```

```
git version 2.1.4
```

1.2 Linux

```
# Instalación en Ubuntu
# Comprobación de que Git no esté instalado
git
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install git
git -version
```

```
git version 2.1.4
```

```
# Opcional. Instalación de todos los subpaquetes complementarios
sudo apt-get install git-all
```

```
# Instalación en Ubuntu mediante PPA
# Comprobación de que Git no esté instalado
git
sudo add-apt-repository ppa:git-core/ppa
```

```
sudo apt-get update
sudo apt-get install git
# Si ya lo tuviéramos instalado solo tendríamos que ejecutar para
# tener la versión de Git actualizada
sudo apt-get upgrade
```

```
# Instalación en CentOS
# Comprobación de que Git no esté instalado
git
sudo yum upgrade
sudo yum install git
git --version
```

```
git version 2.1.4
```

```
# Opcional. Instalación de todos los subpaquetes complementarios
sudo yum install git-all
```

1.3 OS X

```
# Instalación en OS X
# Instalación mediante interfaz gráfica
pwd
ls -la
git --version
```

```
git version 2.1.4
```

```
# Instalación en OS X
# Instalación mediante MacPorts
```

```
sudo port install git-core +svn +doc +bash_completion +gitweb
```

2 Primeros pasos

2.1 Ayuda

```
git
```

```
# Distintas formas de obtener ayuda
git help init
git init --help
man git init
man git-init
```

```
# Lista completa con todos los subcomandos disponibles
git help -a
#  Lista con las guías de conceptos más habituales
git help -g
# Ejemplo de la guía de conceptos  “glossary”
git help glossary
```

2.2 Configuración inicial

```
# Configuro el nombre del usuario a nivel de usuario
git config --global user.name "Fonte Learn"
# Configuro la dirección de correo electrónico
git config --global user.email test@fontelearn.com
# Configuro emacs como editor
git config --global core.editor  emacs
# Configuro notepad.exe como editor en Windows
git config --global core.editor notepad.exe
# Configuro vim como editor
git config --global core.editor vim
# Configuro vimdiff para mostrar las diferencias entre dos archivos
# y como herramienta de resolución de conflictos
git config --global diff.tool vimdiff
git config --global merge.tool vimdiff
# Configuro la salida para que se muestre con colores
git config --global color.ui true
# Configuro las conversiones de final de línea
# Solo necesario en sistemas Windows
git config --global core.autocrlf true
# Configuro que solo se convierta el final de línea en la entrada
git config --global core.autocrlf input
# Muestro la configuración de Git
git config --list
```

```
user.name=Fonte Learn
user.email=test@fontelearn.com
core.editor=vim
merge.tool=vimdiff
color.ui=true
```

3 Ejemplos

3.1 Clonando un repositorio

```
# git clone URL [directorio]
# Accedo a https://github.com/jquery/jquery con el navegador
mkdir ~/proyectos/
cd ~/proyectos/
git clone https://github.com/jquery/jquery.git
```

```
Clonar en «jquery»...
remote: Counting objects: 38320, done.
remote: Compressing objects: 100% (79/79), done.
remote: Total 38320 (delta 44), reused 0 (delta 0), pack-reused 38241
Receiving objects: 100% (38320/38320), 23.17 MiB | 514.00 KiB/s, done.
Resolving deltas: 100% (27073/27073), done.
Checking connectivity... hecho.
```

```
ls
```

```
jquery
```

```
ls jquery -a -1
```

```
.
..
AUTHORS.txt
build
CONTRIBUTING.md
.editorconfig
external
.git
.gitattributes
.gitignore
Gruntfile.js
.jscsrc
.jshintignore
```

```
.jshintrc
LICENSE.txt
.mailmap
.npmignore
package.json
README.md
src
test
.travis.yml
```

```
cd jquery
git log --format='%Cred%h%Creset -%C(yellow)%d%Creset %s
%Cgreen(%cr)' --abbrev-commit --date=relative -10
```

```
47ccf3d - (HEAD, origin/master, origin/HEAD, master) Attributes: do not set
properties to false when removing booleans (hace 23 horas)
15ac848 - Core: make isNumeric limited to strings and numbers (hace 23 horas)
1144e75 - Build: Add "deprecated" to the Testswarm module list (hace 2 días)
e99a3ac - Tests: Add Safari 9 support tests results (hace 4 días)
487d5ca - CSS: Correct misrepresentation of "auto" horizontal margins as 0
(hace 4 días)
c752a50 - Attributes: fix tabIndex on <img> in IE11 (hace 4 días)
3689963 - CSS: use isFinite in place of redundant isNumeric (hace 4 días)
9090d98 - Core: add unit test for isPlainObject(Symbol) (hace 4 días)
32cfc38 - CSS: fix AMD mode for the new showHide module (hace 4 días)
9f9e204 - Offset: allow small differences in offset.top (hace 4 días)
```

```
cd ~/proyectos/
git clone https://github.com/jquery/jquery.git mi_jquery
```

```
Clonar en «mi_jquery»...
remote: Counting objects: 38320, done.
remote: Compressing objects: 100% (79/79), done.
remote: Total 38320 (delta 44), reused 0 (delta 0), pack-reused 38241
Receiving objects: 100% (38320/38320), 23.17 MiB | 516.00 KiB/s, done.
Resolving deltas: 100% (27073/27073), done.
Checking connectivity... hecho.
```

```
ls -1
```

```
jquery
mi_jquery
```

3.2 A partir de un proyecto existente

```
cd ~/proyectos/
# Creo un directorio y accedo a él
mkdir proyecto_existente
cd proyecto_existente
# Creo dos archivos nuevos
touch archivo_a.txt
touch archivo_b.txt
ls -1
```

```
archivo_a.txt
archivo_b.txt
```

```
git init
```

```
Initialized empty Git repository in
/home/usuario/proyectos/proyecto_existente/.git/
```

```
ls -a -1
```

```
archivo_a.txt
archivo_b.txt
.git
```

```
tree .git
```

```
.git
├── branches
├── config
├── description
├── HEAD
```

```
├── hooks
│   ├── applypatch-msg.sample
│   ├── commit-msg.sample
│   ├── post-update.sample
│   ├── pre-applypatch.sample
│   ├── pre-commit.sample
│   ├── prepare-commit-msg.sample
│   ├── pre-push.sample
│   ├── pre-rebase.sample
│   └── update.sample
├── info
│   └── exclude
├── objects
│   ├── info
│   └── pack
└── refs
    ├── heads
    └── tags
```

```
git status
```

```
En la rama master

Commit inicial

Archivos sin seguimiento:
  (use «git add <archivo>...» para incluir lo que se ha de ejecutar)

   archivo_a.txt
   archivo_b.txt

no se ha agregado nada al commit pero existen archivos sin seguimiento (use
«git add» para darle seguimiento)
```

3.3 A partir de un proyecto nuevo

```
# Creo un directorio y accedo a él
mkdir ~/proyectos/proyecto_nuevo/
```

```
cd ~/proyectos/proyecto_nuevo/
git status
```

```
fatal: Not a git repository (or any of the parent directories): .git
```

```
ls -a -1
```

```
.
..
```

```
git init
```

```
Initialized empty Git repository in
/home/usuario/proyectos/proyecto_nuevo/.git/
```

```
ls -a -1
```

```
.
..
.git
```

3.4 Añadiendo un primer archivo

```
touch archivo_a.txt
ls -la
```

```
.
..
archivo_a.txt
.git
```

```
git status
```

```
En la rama master

Commit inicial

Archivos sin seguimiento:
```

```
  (use «git add <archivo>...» para incluir lo que se ha de ejecutar)

    archivo_a.txt

no se ha agregado nada al commit pero existen archivos sin seguimiento (use
«git add» para darle seguimiento)
```

```
git add archivo_a.txt
# git add .
git status
```

```
En la rama master

Commit inicial

Cambios para hacer commit:
  (use «git rm --cached <archivo>...« para eliminar stage)

    new file:   archivo_a.txt
```

3.5 Primer commit

```
git commit -m "Añado el archivo_a.txt vacío"
```

```
[master (root-commit) b4383e5] Añado el archivo_a.txt vacío
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 archivo_a.txt
```

```
git status
```

```
En la rama master
nothing to commit, working directory clean
```

```
# git commit → abre el editor predefinido
```

3.6 Añadiendo más archivos

```
touch archivo_b.txt
```

```
touch archivo_c.txt
git status
```

```
En la rama master
Archivos sin seguimiento:
  (use «git add <archivo>...» para incluir lo que se ha de ejecutar)

   archivo_b.txt
   archivo_c.txt

no se ha agregado nada al commit pero existen archivos sin seguimiento (use
«git add» para darle seguimiento)
```

```
git add archivo_b.txt
git status
```

```
En la rama master
Cambios para hacer commit:
  (use «git reset HEAD <archivo>...«para eliminar stage)

   new file:   archivo_b.txt

Archivos sin seguimiento:
  (use «git add <archivo>...» para incluir lo que se ha de ejecutar)

   archivo_c.txt
```

```
git commit -m "Añado el archivo_b.txt vacío"
```

```
[master d59b4d4] Añado el archivo_b.txt vacío
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 archivo_b.txt
```

```
git status
```

```
En la rama master
Archivos sin seguimiento:
```

```
  (use «git add <archivo>...» para incluir lo que se ha de ejecutar)

   archivo_c.txt

no se ha agregado nada al commit pero existen archivos sin seguimiento (use
«git add» para darle seguimiento)
```

```
git add .
git status
```

```
En la rama master
Cambios para hacer commit:
  (use «git reset HEAD <archivo>...«para eliminar stage)

   new file:   archivo_c.txt
```

```
git commit
# Se abre el editor por defecto (vim)
```

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# En la rama master
# Cambios para hacer commit:
#       new file:   archivo_c.txt
```

Edito el mensaje

```
Añado el archivo_c.txt vacío
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# En la rama master
# Cambios para hacer commit:
#       new file:   archivo_c.txt
```

y guardo el archivo

```
Tecla "Esc"
:wq
```

```
[master 1dea6e3] Añado el archivo_c.txt vacío
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 archivo_c.txt
```

```
# Para ver los commits realizados, uno por línea
git log --oneline
```

```
1dea6e3 Añado el archivo_c.txt vacío
d59b4d4 Añado el archivo_b.txt vacío
b4383e5 Añado el archivo_a.txt vacío
```

3.7 Editando archivos

```
echo "Creo una primera línea en archivo_a.txt" >> archivo_a.txt
git status
```

```
En la rama master
Cambios no preparados para el commit:
  (use «git add <archivo>...» para actualizar lo que se ejecutará)
  (use «git checkout -- <archivo>...« para descartar cambios en le directorio de trabajo)

   modified:   archivo_a.txt

no hay cambios agregados al commit (use «git add» o «git commit -a»)
```

```
git add archivo_a.txt
git status
```

```
En la rama master
Cambios para hacer commit:
  (use «git reset HEAD <archivo>...«para eliminar stage)

   modified:   archivo_a.txt
```

```
echo "Creo una primera línea en el archivo_b.txt" >> archivo_b.txt
echo "Creo una primera línea en el archivo_c.txt" >> archivo_c.txt
```

```
git status
```

```
En la rama master
Cambios para hacer commit:
  (use «git reset HEAD <archivo>...«para eliminar stage)

   modified:   archivo_a.txt

Cambios no preparados para el commit:
  (use «git add <archivo>...» para actualizar lo que se ejecutará)
  (use «git checkout -- <archivo>...« para descartar cambios en le directorio
de trabajo)

   modified:   archivo_b.txt
   modified:   archivo_c.txt
```

```
git add archivo_b.txt
git status
```

```
En la rama master
Cambios para hacer commit:
  (use «git reset HEAD <archivo>...«para eliminar stage)

   modified:   archivo_a.txt
   modified:   archivo_b.txt

Cambios no preparados para el commit:
  (use «git add <archivo>...» para actualizar lo que se ejecutará)
  (use «git checkout -- <archivo>...« para descartar cambios en le directorio
de trabajo)

   modified:   archivo_c.txt
```

```
git commit -m "Introduzco una línea en los archivos archivo_a.txt y
archivo_b.txt"
```

```
[master 4fdd4e4] Introduzco una línea en los archivos archivo_a.txt y
archivo_b.txt
 2 files changed, 2 insertions(+)
```

```
git status
```

```
En la rama master
Cambios no preparados para el commit:
  (use «git add <archivo>...» para actualizar lo que se ejecutará)
  (use «git checkout -- <archivo>...« para descartar cambios en le directorio
de trabajo)

   modified:   archivo_c.txt

no hay cambios agregados al commit (use «git add» o «git commit -a»)
```

```
git log --oneline
```

```
4fdd4e4 Introduzco una línea en los archivos archivo_a.txt y archivo_b.txt
1dea6e3 Añado el archivo_c.txt vacío
d59b4d4 Añado el archivo_b.txt vacío
b4383e5 Añado el archivo_a.txt vacío
```

```
git add archivo_c.txt
git commit -m "Introduzco una línea en el archivo archivo_c.txt"
```

```
[master c51caa1] Introduzco una línea en el archivo archivo_c.txt
 1 file changed, 1 insertion(+)
```

```
git status
```

```
En la rama master
nothing to commit, working directory clean
```

```
git log --oneline
```

```
c51caa1 Introduzco una línea en el archivo archivo_c.txt
4fdd4e4 Introduzco una línea en los archivos archivo_a.txt y archivo_b.txt
1dea6e3 Añado el archivo_c.txt vacío
d59b4d4 Añado el archivo_b.txt vacío
b4383e5 Añado el archivo_a.txt vacío
```

3.8 Borrando archivos

```
touch temporal_1.txt
touch temporal_2.txt
git status
```

```
En la rama master
Archivos sin seguimiento:
  (use «git add <archivo>...» para incluir lo que se ha de ejecutar)

   temporal_1.txt
   temporal_2.txt

no se ha agregado nada al commit pero existen archivos sin seguimiento (use
«git add» para darle seguimiento)
```

```
git add .
git status
```

```
En la rama master
Cambios para hacer commit:
  (use «git reset HEAD <archivo>...«para eliminar stage)

   new file:   temporal_1.txt
   new file:   temporal_2.txt
```

```
git commit -m "Añado dos archivos de prueba para borrar"
```

```
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 temporal_1.txt
 create mode 100644 temporal_2.txt
```

```
rm temporal_1.txt
git status
```

```
En la rama master
Cambios no preparados para el commit:
  (use «git add/rm <archivo>...» para actualizar lo que se ejecutará)
```

```
  (use «git checkout -- <archivo>...« para descartar cambios en le directorio
de trabajo)

   deleted:    temporal_1.txt

no hay cambios agregados al commit (use «git add» o «git commit -a»)
```

```
git rm temporal_1.txt
```

```
rm 'temporal_1.txt'
```

```
git status
```

```
En la rama master
Cambios para hacer commit:
  (use «git reset HEAD <archivo>...«para eliminar stage)

   deleted:    temporal_1.txt
```

```
git commit -m "Borro el archivo temporal_1.txt"
```

```
[master 91b6dda] Borro el archivo temporal_1.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 temporal_1.txt
```

```
git status
```

```
En la rama master
nothing to commit, working directory clean
```

3.9 Borrando archivos (II)

```
git rm temporal_2.txt
```

```
rm 'temporal_2.txt'
```

```
git status
```

```
En la rama master
Cambios para hacer commit:
  (use «git reset HEAD <archivo>...«para eliminar stage)

   deleted:    temporal_2.txt
```

```
git commit -m "Borro el archivo temporal_2.txt"
```

```
[master b5b78cc] Borro el archivo temporal_2.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 temporal_2.txt
```

3.10 Renombrando y moviendo archivos

```
touch temporal_3.txt
touch temporal_4.txt
touch temporal_5.txt
git status
```

```
En la rama master
Archivos sin seguimiento:
  (use «git add <archivo>...» para incluir lo que se ha de ejecutar)

   temporal_3.txt
   temporal_4.txt
   temporal_5.txt

no se ha agregado nada al commit pero existen archivos sin seguimiento (use
«git add» para darle seguimiento)
```

```
# Ponemos los 3 archivos bajo seguimiento
git add .
git status
```

```
En la rama master
Cambios para hacer commit:
  (use «git reset HEAD <archivo>...«para eliminar stage)
```

```
        new file:   temporal_3.txt
        new file:   temporal_4.txt
        new file:   temporal_5.txt
```

```
git commit -m "Añado tres archivos de prueba para moverlos"
```

```
[master 63a453b] Añado tres archivos de prueba para moverlos
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 temporal_3.txt
 create mode 100644 temporal_4.txt
 create mode 100644 temporal_5.txt
```

```
mv temporal_3.txt temporal_3_movido.txt
git status
```

```
En la rama master
Cambios no preparados para el commit:
  (use «git add/rm <archivo>...» para actualizar lo que se ejecutará)
  (use «git checkout -- <archivo>...« para descartar cambios en le directorio de trabajo)

        deleted:    temporal_3.txt

Archivos sin seguimiento:
  (use «git add <archivo>...» para incluir lo que se ha de ejecutar)

        temporal_3_movido.txt

no hay cambios agregados al commit (use «git add» o «git commit -a»)
```

```
git rm temporal_3.txt
```

```
rm 'temporal_3.txt'
```

```
git add temporal_3_movido.txt
git status
```

```
En la rama master
Cambios para hacer commit:
  (use «git reset HEAD <archivo>...«para eliminar stage)

   renamed:    temporal_3.txt -> temporal_3_movido.txt
```

3.11 Renombrando y moviendo archivos (II)

```
git mv temporal_4.txt temporal_4_movido.txt
git status
```

```
En la rama master
Cambios para hacer commit:
  (use «git reset HEAD <archivo>...«para eliminar stage)

   renamed:    temporal_4.txt -> temporal_3_movido.txt
   renamed:    temporal_3.txt -> temporal_4_movido.txt
```

```
mkdir dir_temp
git status
```

```
En la rama master
Cambios para hacer commit:
  (use «git reset HEAD <archivo>...«para eliminar stage)

   renamed:    temporal_4.txt -> temporal_3_movido.txt
   renamed:    temporal_3.txt -> temporal_4_movido.txt
```

```
git mv temporal_5.txt dir_temp/temporal_5.txt
git status
```

```
En la rama master
Cambios para hacer commit:
  (use «git reset HEAD <archivo>...«para eliminar stage)

   renamed:    temporal_5.txt -> dir_temp/temporal_5.txt
   renamed:    temporal_4.txt -> temporal_3_movido.txt
```

```
        renamed:    temporal_3.txt -> temporal_4_movido.txt
```

```
git commit -m "Renombramos tres archivos"
```

```
[master 44a9a8c] Renombramos tres archivos
 3 files changed, 0 insertions(+), 0 deletions(-)
 rename temporal_5.txt => dir_temp/temporal_5.txt (100%)
 rename temporal_4.txt => temporal_3_movido.txt (100%)
 rename temporal_3.txt => temporal_4_movido.txt (100%)
```

3.12 Ignorar archivos

```
touch temporal_6.txt
touch temporal_7.zip
mkdir log
touch log/apache.log
touch log/apache.log1
mkdir imagenes
touch imagenes/fondo.png
touch imagenes/logo.png
mkdir compilados
touch compilados/salida.o
touch compilados/salida.a
touch .gitignore
git status
```

```
En la rama master
Archivos sin seguimiento:
  (use «git add <archivo>...» para incluir lo que se ha de ejecutar)

        .gitignore
        compilados/
        imagenes/
        log/
        temporal_6.txt
        temporal_7.zip
```

```
no se ha agregado nada al commit pero existen archivos sin seguimiento (use
«git add» para darle seguimiento)
```

```
git clean -df
```

```
Eliminando .gitignore
Eliminando compilados/
Eliminando imagenes/
Eliminando log/
Eliminando temporal_6.txt
Eliminando temporal_7.zip
```

```
git status
```

```
En la rama master
nothing to commit, working directory clean
```

```
touch temporal_6.txt
touch temporal_7.zip
mkdir log
touch log/apache.log
touch log/apache.log1
mkdir imagenes
touch imagenes/fondo.png
touch imagenes/logo.png
mkdir compilados
touch compilados/salida.o
touch compilados/salida.a
touch .gitignore
git status
```

```
En la rama master
Archivos sin seguimiento:
  (use «git add <archivo>...» para incluir lo que se ha de ejecutar)
```

```
    .gitignore
    compilados/
    imagenes/
    log/
    temporal_6.txt
    temporal_7.zip

no se ha agregado nada al commit pero existen archivos sin seguimiento (use
«git add» para darle seguimiento)
```

```
# Añadimos el archivo "temporal_6.txt" y el "temporal_7.zip"
echo "temporal_6.txt" >> .gitignore
echo "temporal_7.zip" >> .gitignore
git status
```

```
En la rama master
Archivos sin seguimiento:
  (use «git add <archivo>...» para incluir lo que se ha de ejecutar)

    .gitignore
    compilados/
    imagenes/
    log/

no se ha agregado nada al commit pero existen archivos sin seguimiento (use
«git add» para darle seguimiento)
```

```
# Añadimos todos los archivos .zip y .gz en el directorio raíz
echo "*.zip" >> .gitignore
echo "*.gz" >> .gitignore
git status
```

```
En la rama master
Archivos sin seguimiento:
  (use «git add <archivo>...» para incluir lo que se ha de ejecutar)

    .gitignore
```

```
    compilados/
    imagenes/
    log/

no se ha agregado nada al commit pero existen archivos sin seguimiento (use
«git add» para darle seguimiento)
```

```
# Añadimos los archivos de log que están en el directorio "log"
echo "log/*.log" >> .gitignore
# Añadimos los archivos de rotación de log que están en el
directorio "log"
echo "log/*.log[0-9]" >> .gitignore
git status
```

```
En la rama master
Archivos sin seguimiento:
  (use «git add <archivo>...» para incluir lo que se ha de ejecutar)

    .gitignore
    compilados/
    imagenes/

no se ha agregado nada al commit pero existen archivos sin seguimiento (use
«git add» para darle seguimiento)
```

```
# Añadimos todo el contenido del directorio "imagenes"
echo "imagenes/*" >> .gitignore
git status
```

```
En la rama master
Archivos sin seguimiento:
  (use «git add <archivo>...» para incluir lo que se ha de ejecutar)

    .gitignore
    compilados/

no se ha agregado nada al commit pero existen archivos sin seguimiento (use
«git add» para darle seguimiento)
```

```
# Excepto el archivo "logo.png"
echo "!imagenes/logo.png" >> .gitignore
git status
```

```
En la rama master
Archivos sin seguimiento:
  (use «git add <archivo>...» para incluir lo que se ha de ejecutar)

    .gitignore
    compilados/
    imagenes/

no se ha agregado nada al commit pero existen archivos sin seguimiento (use
«git add» para darle seguimiento)
```

```
# Añadimos los archivos compilados
echo "compilados/*[ao]" >> .gitignore
git status
```

```
En la rama master
Archivos sin seguimiento:
  (use «git add <archivo>...» para incluir lo que se ha de ejecutar)

    .gitignore
    imagenes/

no se ha agregado nada al commit pero existen archivos sin seguimiento (use
«git add» para darle seguimiento)
```

```
git add .
git status
```

```
En la rama master
Cambios para hacer commit:
  (use «git reset HEAD <archivo>...«para eliminar stage)

    new file:   .gitignore
```

```
        new file:   imagenes/logo.png
```

```
git commit -m "Añado el archivo .gitignore y el archivo
imagenes/logo.png"
```

```
 2 files changed, 9 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 imagenes/logo.png
```

```
git status
```

```
En la rama master
nothing to commit, working directory clean
```

3.13 Crear una rama

```
cd ~/proyectos/
mkdir proyecto_ramas
cd proyecto_ramas
git init
```

```
Initialized empty Git repository in
/home/usuario/proyectos/proyecto_ramas/.git/
```

```
touch archivo_a.txt
git add .
git commit -m "Añado el primer archivo"
```

```
[master (root-commit) 015be45] Añado el primer archivo
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 archivo_a.txt
```

```
touch archivo_b.txt
git add .
git commit -m "Añado el segundo archivo"
```

```
[master 3323879] Añado el segundo archivo
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 archivo_b.txt
```

```
git log --oneline
```

```
3323879 Añado el segundo archivo
015be45 Añado el primer archivo
```

```
git branch
```

```
* master
```

```
ls -a -1 .git
```

```
branches
COMMIT_EDITMSG
config
description
HEAD
hooks
index
info
logs
objects
refs
```

```
cat .git/HEAD
```

```
ref: refs/heads/master
```

```
ls -a -1 .git/refs/heads/
```

```
master
```

```
cat .git/refs/heads/master
```

```
33238797baafcb47c385fe20b9df011cf6e6e4f8
```

```
git log --format=format:%H -1
```

```
33238797baafcb47c385fe20b9df011cf6e6e4f8
```

```
git branch pruebas
git branch
```

```
* master
  pruebas
```

```
ls -a -1 .git/refs/heads/
```

```
.
```

```
..
master
pruebas
```

```
cat .git/refs/heads/pruebas
```

```
33238797baafcb47c385fe20b9df011cf6e6e4f8
```

```
cat .git/refs/heads/master
```

```
33238797baafcb47c385fe20b9df011cf6e6e4f8
```

3.14 Mostrar las ramas

```
git branch
```

```
* master
  pruebas
```

```
git branch -r
```

```
git branch -a
```

```
* master
  pruebas
```

3.15 Cambiarse de rama

```
git branch
```

```
* master
  pruebas
```

```
cat .git/HEAD
```

```
ref: refs/heads/master
```

```
git checkout pruebas
```

```
Switched to branch 'pruebas'
```

```
git branch
```

```
  master
* pruebas
```

```
cat .git/HEAD
```

```
ref: refs/heads/pruebas
```

3.16 Trabajando con una rama

```
git checkout pruebas
```

```
Ya está en «pruebas»
```

```
echo "Inserto una línea en el archivo_a.txt" >> archivo_a.txtgit
commit -am "Introduzco una línea en el archivo archivo_a.txt"
```

```
[pruebas 583503e] Introduzco una línea en el archivo archivo_a.txt
 1 file changed, 1 insertion(+)
```

```
git log --oneline
```

```
583503e Introduzco una línea en el archivo archivo_a.txt
3323879 Añado el segundo archivo
015be45 Añado el primer archivo
```

```
cat archivo_a.txt
```

```
Inserto una línea en el archivo_a.txt
```

```
git checkout master
```

```
Switched to branch 'master'
```

```
cat archivo_a.txt
```

```

```

```
git log --oneline
```

```
3323879 Añado el segundo archivo
015be45 Añado el primer archivo
```

```
git checkout pruebas
```

```
Switched to branch 'pruebas'
```

```
git log --oneline
```

```
583503e Introduzco una línea en el archivo archivo_a.txt
3323879 Añado el segundo archivo
015be45 Añado el primer archivo
```

3.17 Crear y cambiar a una rama

```
# git checkout master
# git branch [nombre_de_la_rama]
# git checkout [nombre_de_la_rama]
```

```
git checkout master
```

```
Switched to branch 'master'
```

```
git checkout -b experimento
```

```
Switched to a new branch 'experimento'
```

```
echo "Experimento añadiendo una nueva línea al archivo_a.txt en la
```

```
rama experimento" >> archivo_a.txt
git status
```

```
En la rama experimento
Cambios no preparados para el commit:
  (use «git add <archivo>...» para actualizar lo que se ejecutará)
  (use «git checkout -- <archivo>...« para descartar cambios en le directorio de trabajo)

   modified:   archivo_a.txt

no hay cambios agregados al commit (use «git add» o «git commit -a»)
```

```
git commit -am "Experimentando con una nueva línea en el archivo_a.txt en la rama experimento"
```

```
[experimento bb6104a] Experimentando con una nueva línea en el archivo_a.txt en la rama experimento
 1 file changed, 1 insertion(+)
```

```
git log --oneline
```

```
bb6104a Experimentando con una nueva línea en el archivo_a.txt en la rama experimento
3323879 Añado el segundo archivo
015be45 Añado el primer archivo
```

```
cat archivo_a.txt
```

```
Experimento añadiendo una nueva línea al archivo_a.txt en la rama experimento
```

```
git checkout master
git log --oneline
```

```
3323879 Añado el segundo archivo
015be45 Añado el primer archivo
```

```
cat archivo_a.txt
```

```
git checkout pruebas
git log --oneline
```

```
583503e Introduzco una línea en el archivo archivo_a.txt
3323879 Añado el segundo archivo
015be45 Añado el primer archivo
```

```
cat archivo_a.txt
```

```
Inserto una línea en el archivo_a.txt
```

```
git log --oneline --graph --all --decorate
```

```
* bb6104a (experimento) Experimentando con una nueva línea en el archivo_a.txt
| * 583503e (HEAD, pruebas) Introduzco una línea en el archivo archivo_a.txt
|/
* 3323879 (master) Añado el segundo archivo
* 015be45 Añado el primer archivo
```

3.18 Renombrar una rama

```
# git branch -m [nombre_actual] [nombre_nuevo]
git branch
```

```
  experimento
  master
* pruebas
```

```
git branch -m experimento intento
git branch
```

```
  intento
  master
* pruebas
```

3.19 Borrar una rama

```
# git branch -d [nombre_de_la_rama]
git checkout master
git branch rama_temporal
git branch
```

```
  intento
* master
  pruebas
  rama_temporal
```

```
git checkout rama_temporal
git branch -d rama_temporal
```

```
error: Cannot delete the branch 'rama_temporal' which you are currently on.
```

```
git checkout master
git branch -d rama_temporal
```

```
Deleted branch rama_temporal (was 3323879).
```

```
git branch
```

```
  intento
* master
  pruebas
```

3.20 Fusionar ramas

```
# git merge
git branch
```

```
  intento
* master
  pruebas
```

```
git log --oneline --graph --all --decorate
```

```
* bb6104a (intento) Experimentando con una nueva línea en el archivo_a.txt en
| * 583503e (pruebas) Introduzco una línea en el archivo archivo_a.txt
|/
* 3323879 (HEAD, master) Añado el segundo archivo
* 015be45 Añado el primer archivo
```

```
git checkout master
git merge intento
```

```
Updating 3323879..bb6104a
Fast-forward
 archivo_a.txt | 1 +
 1 file changed, 1 insertion(+)
```

```
git log --oneline --graph --all --decorate
```

```
* bb6104a (HEAD, master, intento) Experimentando con una nueva línea en el
| * 583503e (pruebas) Introduzco una línea en el archivo archivo_a.txt
|/
* 3323879 Añado el segundo archivo
* 015be45 Añado el primer archivo
```

```
git checkout master
cat archivo_a.txt
```

```
Experimento añadiendo una nueva línea al archivo_a.txt en la rama experimento
```

```
git checkout master
echo "Añado una segunda línea al archivo_a.txt en la rama master" >>
archivo_a.txt
git commit -am "Añado una segunda línea al archivo_a.txt en la rama
master"
```

```
[master b831151] Añado una segunda línea al archivo_a.txt en la rama master
 1 file changed, 1 insertion(+)
```

```
git checkout intento

echo "Añado una primera línea al archivo_b.txt en la rama intento"
>> archivo_b.txt

git commit -am "Añado una primera línea al archivo_b.txt en la rama
intento"
```

```
[intento 157b399] Añado una primera línea al archivo_b.txt en la rama intento
 1 file changed, 1 insertion(+)
```

```
git log --oneline --graph --all --decorate
```

```
* 157b399 (HEAD, intento) Añado una primera línea al archivo_b.txt en la rama
| * b831151 (master) Añado una segunda línea al archivo_a.txt en la rama
|/
* bb6104a Experimentando con una nueva línea en el archivo_a.txt en la rama
| * 583503e (pruebas) Introduzco una línea en el archivo archivo_a.txt
|/
* 3323879 Añado el segundo archivo
* 015be45 Añado el primer archivo
```

```
git checkout master

git merge intento
```

```
Merge made by the 'recursive' strategy.
 archivo_b.txt | 1 +
 1 file changed, 1 insertion(+)
```

```
git checkout master

cat archivo_b.txt
```

```
Añado una primera línea al archivo_b.txt en la rama intento
```

```
git log --oneline --graph --all --decorate
```

```
*   2e4da58 (HEAD, master) Merge branch 'intento'
|\
| * 157b399 (intento) Añado una primera línea al archivo_b.txt en la rama
```

```
* | b831151 Añado una segunda línea al archivo_a.txt en la rama master
|/
* bb6104a Experimentando con una nueva línea en el archivo_a.txt en la rama
| * 583503e (pruebas) Introduzco una línea en el archivo archivo_a.txt
|/
* 3323879 Añado el segundo archivo
* 015be45 Añado el primer archivo
```

3.21 Conflictos en la fusión entre ramas

```
git checkout master
cat archivo_a.txt
```

```
Experimento añadiendo una nueva línea al archivo_a.txt en la rama experimento
Añado una segunda línea al archivo_a.txt en la rama master
```

```
git checkout pruebas
cat archivo_a.txt
```

```
Inserto una línea en el archivo_a.txt
```

```
git checkout master
git merge pruebas
```

```
Automezclado archivo_a.txt
CONFLICTO(contenido): conflicto de fusión en archivo_a.txt
Automatic merge failed; fix conflicts and then commit the result.
```

```
git status
```

```
En la rama master
Tiene rutas sin fusionar.
  (solucione los conflictos y ejecute «git commit»)

Rutas no combinadas:
  (use «git add <archivo>...» para marcar resolución)

   modificado por ambos:   archivo_a.txt

no hay cambios agregados al commit (use «git add» o «git commit -a»)
```

```
cat archivo_a.txt
```

```
<<<<<<< HEAD
Experimento añadiendo una nueva línea al archivo_a.txt en la rama experimento
Añado una segunda línea al archivo_a.txt en la rama master
=======
Inserto una línea en el archivo_a.txt
>>>>>>> pruebas
```

```
git merge --abort
git status
```

```
En la rama master
nothing to commit, working directory clean
```

```
cat archivo_a.txt
```

```
Experimento añadiendo una nueva línea al archivo_a.txt en la rama experimento
Añado una segunda línea al archivo_a.txt en la rama master
```

```
# Volvemos a llevar a cabo la fusión
git checkout master
git merge pruebas
```

```
Automezclado archivo_a.txt
CONFLICTO(contenido): conflicto de fusión en archivo_a.txt
Automatic merge failed; fix conflicts and then commit the result.
```

```
cat archivo_a.txt
```

```
<<<<<<< HEAD
Experimento añadiendo una nueva línea al archivo_a.txt en la rama experimento
Añado una segunda línea al archivo_a.txt en la rama master
=======
Inserto una línea en el archivo_a.txt
>>>>>>> pruebas
```

```
# Edito el archivo
echo "Resuelvo el conflicto en esta primera línea del archivo_a.txt"
> archivo_a.txt
echo "También modifico la segunda línea del archivo_a.txt" >>
archivo_a.txt
cat archivo_a.txt
```

```
Resuelvo el conflicto en esta primera línea del archivo_a.txt
También modifico la segunda línea del archivo_a.txt
```

```
git status
```

```
En la rama master
Tiene rutas sin fusionar.
  (solucione los conflictos y ejecute «git commit»)

Rutas no combinadas:
  (use «git add <archivo>...» para marcar resolución)

   modificado por ambos:   archivo_a.txt

no hay cambios agregados al commit (use «git add» o «git commit -a»)
```

```
git add archivo_a.txt
```

```
git status
```

```
En la rama master
Todos los conflictos solucionados pero aún se está fusionando
  (use «git commit» para concluir la fusión)

Cambios para hacer commit:

   modified:   archivo_a.txt
```

```
git commit -m "Resuelto el conflicto en la línea 1 en el
archivo_a.txt"
```

```
[master 6f46ed0] Resuelto el conflicto en la línea 1 en el archivo_a.txt
```

```
git log --oneline --graph --all --decorate -6
```

```
*   6f46ed0 (HEAD, master) Resuelto el conflicto en la línea 1 en el
|\
| * 583503e (pruebas) Introduzco una línea en el archivo archivo_a.txt
* |   2e4da58 Merge branch 'intento'
|\ \
| * | 157b399 (intento) Añado una primera línea al archivo_b.txt en la rama
* | | b831151 Añado una segunda línea al archivo_a.txt en la rama master
|/ /
* | bb6104a Experimentando con una nueva línea en el archivo_a.txt en la rama
|/
```

3.22 Rebase

```
mkdir ~/proyectos/proyecto_rebase
cd ~/proyectos/proyecto_rebase
git init
```

```
Initialized empty Git repository in
/home/usuario/proyectos/proyecto_rebase/.git/
```

```
touch archivo_1.txt
```

```
git add .
git commit -m "Commit inicial"
```

```
git checkout -b rama_1
echo "Línea 1" >> archivo_1.txt
git commit -am "Añado una línea"
```

```
git checkout master
echo "Línea 1" >> archivo_2.txt
git add .
git commit -m "Añado un nuevo archivo"
```

```
git merge rama_1
```

```
git log --oneline --graph --all --decorate
```

```
*   c38a122 (HEAD, master) Merge branch 'rama_1'
|\
| * ce52c8a (rama_1) Añado una línea
* | 7f5ba8e Añado un nuevo archivo
|/
* d6b1012 Commit inicial
```

3.23 Rebase (II)

```
mkdir ~/proyectos/proyecto_rebase_2
cd ~/proyectos/proyecto_rebase_2
git init
```

```
touch archivo_1.txt
git add .
git commit -m "Commit inicial"
```

```
git checkout -b rama_1
echo "Línea 1" >> archivo_1.txt
git commit -am "Añado una línea"
```

```
echo "Línea 2" >> archivo_1.txt
git commit -am "Añado una segunda línea"
```

```
echo "Línea 3" >> archivo_1.txt
git commit -am "Añado una tercera línea"
```

```
git checkout master
echo "Línea 1" >> archivo_2.txt
git add .
git commit -m "Añado un nuevo archivo"
git log --oneline --graph --all --decorate
```

```
* 697dab8 (HEAD, master) Añado un nuevo archivo
| * f81da58 (rama_1) Añado una tercera línea
| * 2a65ce1 Añado una segunda línea
| * 9241aad Añado una línea
|/
* acede8b Commit inicial
```

```
git checkout rama_1
git rebase master
```

```
First, rewinding head to replay your work on top of it...
Applying: Añado una línea
Applying: Añado una segunda línea
Applying: Añado una tercera línea
```

```
git log --oneline --graph --all --decorate
```

```
* c7db69d (HEAD, rama_1) Añado una tercera línea
* 479d3ea Añado una segunda línea
* 7178441 Añado una línea
* 697dab8 (master) Añado un nuevo archivo
* acede8b Commit inicial
```

```
git checkout master
git merge rama_1
```

```
Updating 697dab8..c7db69d
Fast-forward
 archivo_1.txt | 3 +++
 1 file changed, 3 insertions(+)
```

```
git log --oneline --graph --all --decorate
```

```
* c7db69d (HEAD, rama_1, master) Añado una tercera línea
* 479d3ea Añado una segunda línea
* 7178441 Añado una línea
* 697dab8 Añado un nuevo archivo
* acede8b Commit inicial
```

3.24 Stash

```
mkdir ~/proyectos/proyecto_stash
cd ~/proyectos/proyecto_stash
git init
git branch rama_1
```

```
fatal: El nombre del objeto no es válido: «master».
```

```
echo "Primera línea del archivo_1.txt" >> archivo_1.txt
git add .
git commit -m "Commit inicial"
echo "Segunda línea del archivo_1.txt" >> archivo_1.txt
echo "Primera línea del archivo_2.txt" >> archivo_2.txt
ls -a -1
```

```
archivo_1.txt
archivo_2.txt
.git
```

```
# Supongamos que tenemos una interrupción
git status
```

```
En la rama master
Cambios no preparados para el commit:
  (use «git add <archivo>...» para actualizar lo que se ejecutará)
  (use «git checkout -- <archivo>...« para descartar cambios en le directorio de trabajo)

   modified:   archivo_1.txt

Archivos sin seguimiento:
  (use «git add <archivo>...» para incluir lo que se ha de ejecutar)

   archivo_2.txt

no hay cambios agregados al commit (use «git add» o «git commit -a»)
```

```
# Tenemos un archivo con cambios y otro sin estar bajo seguimiento
git add archivo_2.txt
git status
```

```
En la rama master
Cambios para hacer commit:
  (use «git reset HEAD <archivo>...«para eliminar stage)
```

```
    new file:   archivo_2.txt


Cambios no preparados para el commit:
  (use «git add <archivo>...» para actualizar lo que se ejecutará)
  (use «git checkout -- <archivo>...« para descartar cambios en le directorio
de trabajo)


    modified:   archivo_1.txt
```

```
cat archivo_1.txt
```

```
Primera línea del archivo_1.txt
Segunda línea del archivo_1.txt
```

```
cat archivo_2.txt
```

```
Primera línea del archivo_2.txt
```

```
# Almacenamos estos cambios en el área de stash
git stash
```

```
Saved working directory and index state WIP on master: 7644181 Commit inicial
HEAD is now at 7644181 Commit inicial
```

```
ls -a -1
```

```
archivo_1.txt
.git
```

```
cat archivo_1.txt
```

```
Primera línea del archivo_1.txt
```

```
git stash
```

```
No hay cambios locales que guardar
```

```
git stash list
```

```
stash@{0}: WIP on master: 7644181 Commit inicial
```

```
git stash show stash@{0}
```

```
 archivo_1.txt | 1 +
 archivo_2.txt | 1 +
 2 files changed, 2 insertions(+)
```

```
git stash show stash@{0} -p
```

```
diff --git a/archivo_1.txt b/archivo_1.txt
index 34c5d11..19265b1 100644
--- a/archivo_1.txt
+++ b/archivo_1.txt
@@ -1 +1,2 @@
 Primera línea del archivo_1.txt
+Segunda línea del archivo_1.txt
diff --git a/archivo_2.txt b/archivo_2.txt
new file mode 100644
index 0000000..b054e56
--- /dev/null
+++ b/archivo_2.txt
@@ -0,0 +1 @@
+Primera línea del archivo_2.txt
```

```
git checkout -b rama_1
ls -a -1
```

```
archivo_1.txt
.git
```

```
git stash apply stash@{0}
```

```
En la rama rama_1
Cambios para hacer commit:
```

```
    (use «git reset HEAD <archivo>...«para eliminar stage)

     new file:   archivo_2.txt

Cambios no preparados para el commit:
    (use «git add <archivo>...» para actualizar lo que se ejecutará)
    (use «git checkout -- <archivo>...« para descartar cambios en le directorio
de trabajo)

     modified:   archivo_1.txt
```

```
git stash list
```

```
stash@{0}: WIP on master: 7644181 Commit inicial
```

```
git stash drop stash@{0}
```

```
Dropped stash@{0} (34cabdb17aa8437df93dad59f6b057013aaca910)
```

```
git stash list
```

4 Servidor

4.1 Usuario 1

```
mkdir ~/proyectos/proyecto_compartido
cd ~/proyectos/proyecto_compartido
git init
touch index.html
echo "Simulación del archivo index.html" >> index.html
touch style.css
echo "Simulación del archivo style.css" >> style.css
git add .
git commit -m "Commit inicial. Añado el index.html y el style.css"
```

```
[master (root-commit) 2f8837b] Commit inicial. Añado el index.html y el
style.css
 2 files changed, 2 insertions(+)
 create mode 100644 index.html
 create mode 100644 style.css
```

```
git tag -a v0.0.1 -m "Commit inicial"
git checkout -b dev
touch todo.txt
touch .gitignore
echo "todo.txt" >> .gitignore
git add .gitignore
git commit -m "Añado el archivo .gitignore"
```

```
[dev 1d259a5] Añado el archivo .gitignore
 1 file changed, 1 insertion(+)
 create mode 100644 .gitignore
```

```
git remote add origin
https://amieiro@bitbucket.org/amieiro/proyecto-compartido.git
// git remote add origin git@bitbucket.org:amieiro/proyecto-
compartido.git
```

```
git push -u origin master
```

```
Password for 'https://amieiro@bitbucket.org':
Counting objects: 4, done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 358 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://amieiro@bitbucket.org/amieiro/proyecto-compartido.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

```
git branch -a
```

```
* dev
  master
  remotes/origin/master
```

```
git push -u origin dev
```

```
Password for 'https://amieiro@bitbucket.org':
Counting objects: 3, done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 337 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create pull request for dev:
remote:   https://bitbucket.org/amieiro/proyecto-compartido/pull-requests/new?
source=dev&t=1
remote:
To https://amieiro@bitbucket.org/amieiro/proyecto-compartido.git
 * [new branch]      dev -> dev
Branch dev set up to track remote branch dev from origin.
```

```
git branch -a
```

```
* dev
  master
  remotes/origin/dev
```

```
    remotes/origin/master
```

```
git push --tags
```

```
Password for 'https://amieiro@bitbucket.org':
Counting objects: 1, done.
Writing objects: 100% (1/1), 168 bytes | 0 bytes/s, done.
Total 1 (delta 0), reused 0 (delta 0)
To https://amieiro@bitbucket.org/amieiro/proyecto-compartido.git
 * [new tag]         v0.0.1 -> v0.0.1
```

4.2 Usuario 2

```
adduser jesusamieiro
sudo su jesusamieiro
```

```
git config --global user.email "jesus@jesusamieiro.com"
git config --global user.name "Jesús Amieiro"
cd ~
pwd
mkdir ~/proyectos
cd ~/proyectos
git clone https://JesusAmieiro@bitbucket.org/amieiro/proyecto-
compartido.git
```

```
Clonar en «proyecto-compartido»...
Password for 'https://JesusAmieiro@bitbucket.org':
remote: Counting objects: 8, done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 8 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (8/8), done.
Checking connectivity... hecho.
```

```
cd proyecto-compartido
git branch -a -v
```

```
* master                2f8837b Commit inicial. Añado el index.html y el
style.css
  remotes/origin/HEAD   -> origin/master
  remotes/origin/dev    1d259a5 Añado el archivo .gitignore
  remotes/origin/master 2f8837b Commit inicial. Añado el index.html y el
style.css
```

git checkout -b dev origin/dev

```
Branch dev set up to track remote branch dev from origin.
Switched to a new branch 'dev'
```

```
git branch -a -v
```

```
* dev                   1d259a5 Añado el archivo .gitignore
  master                2f8837b Commit inicial. Añado el index.html y el
style.css
  remotes/origin/HEAD   -> origin/master
  remotes/origin/dev    1d259a5 Añado el archivo .gitignore
  remotes/origin/master 2f8837b Commit inicial. Añado el index.html y el
style.css
```

```
mkdir img
cd img/
touch logo.png
touch logo.xcf
cd ..
echo "img/*.png" >> .gitignore
git status
```

```
En la rama dev
Su rama está actualizada con «origin/dev».
Cambios no preparados para el commit:
  (use «git add <archivo>...» para actualizar lo que se confirmará)
  (use «git checkout -- <archivo>...» para descartar cambios en el directorio
de trabajo)

   modificado:    .gitignore
```

```
Archivos sin seguimiento:
  (use «git add <archivo>...» para incluir en lo que se ha de confirmar)

	img/

no hay cambios agregados al commit (use «git add» o «git commit -a»)
```

```
git add .gitignore
git add img/logo.xcf
git status
```

```
En la rama dev
Su rama está actualizada con «origin/dev».
Cambios para hacer commit:
  (use «git reset HEAD <archivo>...» para sacar del stage)

	modificado:    .gitignore
	nuevo archivo: img/logo.xcf
```

```
git commit -m "Añado el logo y modifico el .gitignore"
```

```
[dev 5ae9a02] Añado el logo y modifico el .gitignore
 2 files changed, 1 insertion(+)
 create mode 100644 img/logo.xcf
```

```
git checkout master
```

```
Switched to branch 'master'
Su rama está actualizada con «origin/master».
```

```
git merge dev
```

```
Updating 2f8837b..5ae9a02
Fast-forward
 .gitignore   | 2 ++
 img/logo.xcf | 0
 2 files changed, 2 insertions(+)
```

```
 create mode 100644 .gitignore
 create mode 100644 img/logo.xcf
```

```
git push -all
```

```
Password for 'https://JesusAmieiro@bitbucket.org':
Counting objects: 5, done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (5/5), 443 bytes | 0 bytes/s, done.
Total 5 (delta 0), reused 0 (delta 0)
To https://JesusAmieiro@bitbucket.org/amieiro/proyecto-compartido.git
   1d259a5..5ae9a02  dev -> dev
   2f8837b..5ae9a02  master -> master
```

4.3 Usuario 1

```
git checkout master
```

```
Switched to branch 'master'
Su rama está actualizada con «origin/master».
```

```
git pull --all
```

```
Fetching origin
Password for 'https://amieiro@bitbucket.org':
remote: Counting objects: 5, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 5 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (5/5), done.
De https://bitbucket.org/amieiro/proyecto-compartido
   2f8837b..5ae9a02  master     -> origin/master
   1d259a5..5ae9a02  dev        -> origin/dev
Updating 2f8837b..5ae9a02
Fast-forward
 .gitignore   | 2 ++
 img/logo.xcf | 0
 2 files changed, 2 insertions(+)
 create mode 100644 .gitignore
```

```
 create mode 100644 img/logo.xcf
```

```
cat .gitignore
```

```
todo.txt
img/*.png
```

```
git branch -a -v
```

```
  dev                   1d259a5 [behind 1] Añado el archivo .gitignore
* master                5ae9a02 Añado el logo y modifico el .gitignore
  remotes/origin/dev    5ae9a02 Añado el logo y modifico el .gitignore
  remotes/origin/master 5ae9a02 Añado el logo y modifico el .gitignore
```

```
git checkout dev
```

```
Switched to branch 'dev'
Your branch is behind 'origin/dev' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)
```

```
git pull
```

```
Password for 'https://amieiro@bitbucket.org':
Updating 1d259a5..5ae9a02
Fast-forward
 .gitignore   | 1 +
 img/logo.xcf | 0
 2 files changed, 1 insertion(+)
 create mode 100644 img/logo.xcf
```

```
git branch -a -v
```

```
* dev                   5ae9a02 Añado el logo y modifico el .gitignore
  master                5ae9a02 Añado el logo y modifico el .gitignore
  remotes/origin/dev    5ae9a02 Añado el logo y modifico el .gitignore
  remotes/origin/master 5ae9a02 Añado el logo y modifico el .gitignore
```

```
mkdir js/
cd js/
touch jquery-2.1.0.min.js
cd ..
git add .
git commit -m "Añado el archivo jquery-2.1.0.min.js"
```

```
[dev 5a30a8f] Añado el archivo jquery-2.1.0.min.js
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 js/jquery-2.1.0.min.js
```

```
git checkout master
```

```
Switched to branch 'master'
Su rama está actualizada con «origin/master».
```

```
git merge dev
```

```
Updating 5ae9a02..5a30a8f
Fast-forward
 js/jquery-2.1.0.min.js | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 js/jquery-2.1.0.min.js
```

```
git push --all
```

```
Password for 'https://amieiro@bitbucket.org':
Counting objects: 3, done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 339 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To https://amieiro@bitbucket.org/amieiro/proyecto-compartido.git
   5ae9a02..5a30a8f  dev -> dev
   5ae9a02..5a30a8f  master -> master
```

4.4 Usuario 2

```
git fetch
```

```
Password for 'https://JesusAmieiro@bitbucket.org':
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
De https://bitbucket.org/amieiro/proyecto-compartido
   5ae9a02..5a30a8f  master     -> origin/master
   5ae9a02..5a30a8f  dev        -> origin/dev
```

```
git branch -a -v
```

```
  dev                   5ae9a02 [behind 1] Añado el logo y modifico el
.gitignore
* master                5ae9a02 [behind 1] Añado el logo y modifico el
.gitignore
  remotes/origin/HEAD   -> origin/master
  remotes/origin/dev    5a30a8f Añado el archivo jquery-2.1.0.min.js
  remotes/origin/master 5a30a8f Añado el archivo jquery-2.1.0.min.js
```

```
git merge origin/master
```

```
Updating 5ae9a02..5a30a8f
Fast-forward
 js/jquery-2.1.0.min.js | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 js/jquery-2.1.0.min.js
```

```
git branch -a -v
```

```
  dev                   5ae9a02 [behind 1] Añado el logo y modifico el
.gitignore
* master                5a30a8f Añado el archivo jquery-2.1.0.min.js
  remotes/origin/HEAD   -> origin/master
  remotes/origin/dev    5a30a8f Añado el archivo jquery-2.1.0.min.js
  remotes/origin/master 5a30a8f Añado el archivo jquery-2.1.0.min.js
```

```
git checkout dev
```

```
git checkout dev
git merge origin/dev
```

```
git merge origin/dev
```

```
Updating 5ae9a02..5a30a8f
Fast-forward
 js/jquery-2.1.0.min.js | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 js/jquery-2.1.0.min.js
```